

#2

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:)
)
Takahiro MASUDA, et al.)
) Group Art Unit: Unassigned
Serial No.: To be assigned)
) Examiner: Unassigned
Filed: January 19, 2001)
)
For: **INSTALLATION METHOD,**)
 ACTIVATION METHOD,)
 EXECUTION APPARATUS)
 AND MEDIUM OF)
 APPLICATION PROGRAM)

jc972 U.S. PTO
09/764352
01/19/01

SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN
APPLICATION IN ACCORDANCE
WITH THE REQUIREMENTS OF 37 C.F.R. §1.55

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

In accordance with the provisions of 37 C.F.R. §1.55, the applicants submit herewith a certified copy of the following foreign application:

Japanese Patent Application No. 2000-115239
Filed: April 17, 2000.

It is respectfully requested that the applicants be given the benefit of the foreign filing date as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. §119.

Respectfully submitted,
STAAS & HALSEY LLP

By: _____
James D. Halsey, Jr.
Registration No. 22,729

700 11th Street, N.W., Ste. 500
Washington, D.C. 20001
(202) 434-1500

Date: January 19, 2001

日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT

091082



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日

Date of Application:

2000年 4月17日

出願番号

Application Number:

特願2000-115239

出願人

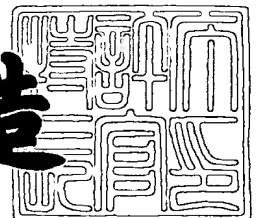
Applicant(s):

富士通株式会社

2000年 9月 1日

特許庁長官
Commissioner,
Patent Office

及川耕造



出証番号 出証特2000-3070886

【書類名】 特許願

【整理番号】 0000021

【提出日】 平成12年 4月17日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 13/00

【発明の名称】 アプリケーションのインストール方法、起動方法、実行装置および媒体

【請求項の数】 12

【発明者】

 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

 【氏名】 増田 高弘

【発明者】

 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

 【氏名】 外川 好房

【特許出願人】

 【識別番号】 000005223

 【氏名又は名称】 富士通株式会社

【代理人】

 【識別番号】 100089244

 【弁理士】

 【氏名又は名称】 遠山 勉

【選任した代理人】

 【識別番号】 100090516

 【弁理士】

 【氏名又は名称】 松倉 秀実

 【連絡先】 03-3669-6571

【手数料の表示】

【予納台帳番号】 012092

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9705606

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 アプリケーションのインストール方法、起動方法、実行装置および媒体

【特許請求の範囲】

【請求項 1】 データファイルを解析して特徴点を抽出するステップと、
抽出された特徴点に基づいて当該データファイルに適したアプリケーションのバージョンを決定するステップと、

決定されたバージョンのアプリケーションプログラムが既にインストールされているか否かを判定するステップと、

インストールされていないときに当該バージョンのアプリケーションプログラムを新たにインストールするステップとからなるアプリケーションのインストール方法。

【請求項 2】 前記特徴点はデータファイルに含まれるマクロ命令の予約語であり、バージョン毎に特有な予約語を検出することにより前記バージョンを決定する請求項 1 記載のアプリケーションのインストール方法。

【請求項 3】 前記特徴点はデータファイルに含まれるマクロ命令の構文パターンであり、バージョン毎に特有な構文パターンを検出することにより前記バージョンを決定する請求項 1 記載のアプリケーションのインストール方法。

【請求項 4】 2 以上のデータファイルを解析してそれぞれの特徴点を抽出するステップと、

前記特徴点に基づいて前記データファイルを可読可能なアプリケーションプログラムのバージョンを決定するステップと、

前記データファイルと、前記で決定されたバージョンのアプリケーションプログラムとを同時に表示するステップと、

表示されたアプリケーションプログラムを前記いずれかのデータファイルと関連付けて起動するステップとからなるアプリケーションの起動方法。

【請求項 5】 前記データファイルは、前記アプリケーションプログラムの対応バージョン毎に異なるシンボル図形で表示される請求項 3 記載のアプリケーションの起動方法。

【請求項6】 前記アプリケーションプログラムが選択されたときに、当該アプリケーションプログラムのバージョンに対応したデータファイルのみが表示される請求項3記載のアプリケーションの起動方法。

【請求項7】 前記データファイルに対して対応するアプリケーションプログラムが存在しないときに、対応するアプリケーションプログラムのインストールを実行する請求項3記載のアプリケーションの起動方法。

【請求項8】 前記アプリケーションプログラムのインストールを実行する際に空き領域が不足している場合に既存のファイルを削除するステップを含む請求項3記載のアプリケーションの起動方法。

【請求項9】 2以上のデータファイルを解析してそれぞれの特徴点を抽出するステップと、

前記特徴点に基づいて前記データファイルを可読可能なアプリケーションプログラムのバージョンを決定するステップと、

前記データファイルと、前記で決定されたバージョンのアプリケーションプログラムとを同時に表示するステップと、

表示されたアプリケーションプログラムを前記いずれかのデータファイルと関連付けて起動するステップとを含むプログラムを記録した記録媒体。

【請求項10】 2以上のデータファイルを解析してそれぞれの特徴点を抽出する手段と、

前記特徴点に基づいて前記データファイルを可読可能なアプリケーションプログラムのバージョンを決定する手段と、

前記データファイルと、前記で決定されたバージョンのアプリケーションプログラムとを同時に表示する手段と、

表示されたアプリケーションプログラムを前記いずれかのデータファイルと関連付けて起動する手段とからなるアプリケーションプログラム実行装置。

【請求項11】 データファイルを解析して特徴点を抽出する手段と、

抽出された特徴点に基づいて当該データファイルに適したアプリケーションのバージョンを決定する手段と、

決定されたバージョンのアプリケーションプログラムが既にインストールされ

ているか否かを判定する手段と、

インストールされていないときに当該バージョンのアプリケーションプログラムを新たにインストールする手段とからなるアプリケーションインストール装置

。

【請求項 1 2】 データファイルを解析して特徴点を抽出するステップと、

抽出された特徴点に基づいて当該データファイルに適したアプリケーションのバージョンを決定するステップと、

決定されたバージョンのアプリケーションプログラムが既にインストールされているか否かを判定するステップと、

インストールされていないときに当該バージョンのアプリケーションプログラムを新たにインストールするステップとを順次実行するプログラムを記憶した記憶媒体。

【発明の詳細な説明】

【 0 0 0 1 】

【発明の属する技術分野】

本発明はデータに必要なバージョンのアプリケーションの自動インストールおよび自動起動の技術に関する。

【 0 0 0 2 】

【従来の技術】

パーソナルコンピュータやワードプロセッサ専用機で作成された文書ファイルやスプレッドシートファイルのデータは、これを作成・更新するアプリケーションプログラムのバージョンアップにしたがってデータ形式が変更されるため、新しいバージョンのプログラムで作成されたデータが旧バージョンのプログラムでは利用できないことが多い。逆に新しいバージョンのアプリケーションを使っている場合でも、旧バージョンのアプリケーションで作成されたデータが編集できないことがある。

【 0 0 0 3 】

ワードプロセッサのアプリケーションの場合、アプリケーション側のバージョンによってはデータ格納形式が異なるため、文書のレイアウトが別のバージョン

のアプリケーションプログラムでは再現できない場合がある。そのために、文書コンバータと呼ばれる別のアプリケーションを介在させてデータ形式を変換させる必要があった。

【0004】

また、スプレッドシートと呼ばれる表計算プログラムの場合、バージョンによって使用可能なオブジェクトが異なるため、異なるバージョンで作成されたデータのあるプログラムで読み出そうとした場合、エラーとなることがある。

【0005】

さらに、この種のプログラムではマクロと呼ばれる一定の手順を記述した簡易プログラムが付属しているが、このマクロの仕様、たとえば予約語の有無などもプログラムのバージョンによって異なるため、作成したバージョンと異なるバージョンのプログラムでデータファイルを開いた場合に当該マクロが実行不可能となりエラーを生じる可能性があった。

【0006】

インターネットで代表されるネットワーク社会では、通信を経由してこの種のデータファイルがやりとりされることが多く、受信したデータファイルが、自分のコンピュータにインストールされているアプリケーションプログラムで利用できるバージョンであるか否かは重要な問題だった。

【0007】

したがって、あるユーザーがインストールされているアプリケーションのバージョンでそのデータファイルが扱えない場合には、ユーザーは利用可能なアプリケーションのバージョンを判定して、そのアプリケーションの入手先を調査し、入手後に自分のコンピュータ上にインストールする必要があった。

【0008】

【発明が解決しようとする課題】

ところで、バージョンによって異なる拡張子が付与されるデータファイルであれば拡張子を参照するだけで利用可能なアプリケーションプログラムのバージョンを知ることができる。

【0009】

しかし、docなどの文書形式のようにバージョンが異なっても統一した拡張子が付与されるアプリケーションプログラムの場合には、データファイルを外部から観察しただけでは適切なバージョンを判定することは困難だった。そのため、最悪の場合にはユーザーは種々のバージョンのアプリケーションプログラムを試行的にインストールしてみてデータファイルのオープンを繰り返してみる必要があった。

【 0 0 1 0 】

本発明は、このような点に鑑みてなされたものであり、ユーザーにわずらわしいバージョンの判定を行わせることなく、データファイルに対して最適なバージョンのアプリケーションプログラムを自動インストールし、当該アプリケーションの自動起動を可能にすることを技術的課題とする。

【 0 0 1 1 】

【課題を解決するための手段】

本発明は、まず読み出そうとするデータファイルを解析して特徴点を抽出し、抽出された特徴点によってそのデータファイルに適したアプリケーションのバージョンを識別する。そして、前記で識別されたバージョンのアプリケーションプログラムが既にインストールされているか否かを判定し、この判定結果によりインストールされていないときには当該バージョンのアプリケーションプログラムを新たにインストールするようにした。

【 0 0 1 2 】

この特徴点とは、たとえば文書ファイルにおけるマクロ命令の予約語や構文パターンである。これらの予約語や構文パターンはそれを作成したアプリケーションプログラムのバージョンに依存していることが多く、これらを抽出することによって当該文書ファイルを開くための最適なアプリケーションプログラムのバージョンを知ることができる。

【 0 0 1 3 】

また、バージョンの解析結果に基づいて文書ファイルと対応するバージョンのアプリケーションプログラムとを同時に表示装置上に表示し、いずれのバージョンのアプリケーションプログラムで文書ファイルを開くかをユーザーに選択させ

るようにしてもよい。この場合、解析結果で得られた情報に基づいて文書ファイルのバージョン毎に異なるアイコンを付記してもよいし、アプリケーションプログラムをまず選択させた場合、このアプリケーションプログラムで開くことのできる文書ファイルだけを表示させるようにしてもよい。

【 0 0 1 4 】

【発明の実施の形態】

【 0 0 1 5 】

【実施例 1】

以下、図面に基づいて、本発明の実施の形態を説明する。

図 1 は、本発明の実施例による自動インストールシステムの機能ブロック図、図 2 は本実施例のシステム構成を示すブロック図である。

【 0 0 1 6 】

図 2 に示すように、ネットワーク 3 0 を介してサーバ 3 1 と端末装置 2 1 とが接続されている。端末装置 2 1 はそれ自体でコンピュータシステムを構成しており、本実施例の自動インストール処理は端末装置 2 1 だけでも実施可能である。

【 0 0 1 7 】

端末装置 2 1 は、バス 2 2 を介して中央処理装置（CPU）2 3、メモリ 2 4、表示装置（CRT）2 5、印刷装置（PRT）2 6 およびハードディスク装置（HD）2 7 を有している。また、外部補助入力手段としてマウス 2 9 を有している。また、前記バス 2 2 は、通信インターフェース 2 8 を介して前記ネットワーク 3 0 と接続されている。

【 0 0 1 8 】

図 1 は、本実施例の自動インストール手順を示しており、かかる機能はハードディスク装置（HD）2 7 に格納されたプログラムがメモリ 2 4 に読み込まれて、さらに該プログラムを中央処理装置（CPU）2 3 が順次実行することにより実現される。

【 0 0 1 9 】

ターゲット文書 1 は、たとえばワードプロセッサ用の文書ファイルやスプレッドシート用のデータファイルであり、ハードディスク（HD）2 7 内に登録され

ている。ターゲット文書 1 は、通常の場合、文書領域とマクロ記述領域とを有しており、所定のワードプロセッサプログラムで当該ターゲット文書 1 を読み込んだ場合に、その文書に付随したマクロ命令で規定された文字列の処理や整形が可能となっている。

【 0 0 2 0 】

このようなターゲット文書 1 がメモリ 2 4 に読み込まれると、中央処理装置（CPU）2 3 はこのターゲット文書 1 に対して構文解析処理 2 を行う。この構文解析処理は、具体的には、マクロ命令予約語を抽出する作業である。

【 0 0 2 1 】

マクロ命令予約語の抽出は、ターゲット文書 1 から予約語リスト 1 2 を生成することにより行われる。これは具体的には、ハードディスク装置（HD）2 7 に登録された全予約語データベース（DB）1 3 を参照しパターンマッチングによりターゲット文書 1 の中から予約語を抽出し、これを予約語リスト 1 2 に登録することである。そして、この予約語リスト 1 2 とハードディスク装置（HD）2 7 内に登録された全予約バージョン対応リスト 4 とが比較されて（比較参照処理 3）、当該ターゲット文書 1 に対する有効バージョンリスト 6 が生成される。

【 0 0 2 2 】

また、前述の構文解析処理 2 と並行して、中央処理装置（CPU）2 3 は構文パターンの検出を行う（ステップ 5）。この構文パターンの検出では、ターゲット文書 1 中のマクロプログラムをトークンに分割してそらのトークンの属性を判定した上で特定の属性のトークンの結合状態が存在するか否かを判定することにより実現する。たとえば、「Collection-Classのインスタンスとなる変数」＋「(」＋「Strings-Classのインスタンスとなる変数」＋「)」というような結合状態を検出することである。有効バージョンリスト 6 a の生成に際しては、この構文パターンの検出結果も反映させて最終的な有効バージョンリスト 1 4 を生成する。このとき、バージョンリスト 6 a の生成に際しては、マクロ命令予約語の抽出結果と構文パターンの検出結果とのいずれを優先させてもよい。たとえば、マクロ命令予約語の抽出結果に基づくバージョンと、構文パターンの検出結果に基づくバージョンとが異なるときには、より数値の大きい（最新のバージョン）

を優先的に採用して有効バージョンリスト 6 a を生成してもよい。

【 0 0 2 3 】

次に、前記有効バージョンリスト 6 a と、インストール済みバージョンリスト 7 とが比較される（比較処理 8）。

この比較処理 8 の結果、該当するバージョンのアプリケーションプログラムが既に端末装置 2 1 内のハードディスク装置（H D）2 7 内に実行可能な状態でインストールされている場合には当該アプリケーションプログラムを実行し（実行処理 1 0）、インストールされていない場合には、インストール処理 9 を行う。

【 0 0 2 4 】

なお、有効バージョンリスト 6（6 a）の生成の際にバージョンリスト内が空である場合、すなわち有効バージョンが判定できない場合には、警告処理 1 1 がなされる。この警告処理 1 1 は、たとえば表示装置（C R T）2 5 上に視覚的な警告表示を行うことが考えられる。

【 0 0 2 5 】

【実施例 2】

実施例 1 では、比較処理 8 において、中央処理装置（C P U）2 3 が比較処理 8 を実行し、最適なアプリケーションプログラムのバージョンを自動決定する方を説明したが、本実施例 2 では、比較処理 8 の結果を表示装置（C R T）2 5 上でユーザーが一見して把握可能な形式で表示し、ユーザー自身に複数のバージョンのアプリケーションプログラムを選択させる余地を有している点が特徴である。

【 0 0 2 6 】

なお、図 1 で説明した処理の中で、比較処理 8 以前の処理については実施例 1 と同様であるので説明を省略する。

中央処理装置（C P U）2 3 は、ターゲット文書 1 から得られた有効バージョンリスト 6（6 a）とインストール済みバージョンリスト 7 とを比較する際に、表示装置（C R T）2 5 上に図 4 に示すような比較画面 4 1 を表示する。この比較画面 4 1 は、同図左側のアプリケーションリスト表示窓 4 2 と、モジュールリ

スト表示窓43との分割されている。アプリケーションリスト表示窓42は、アプリケーションプログラムがグループ毎に階層的に表示可能となっており、同図において、スプレッドシート（表計算プログラム）のフォルダ内にバージョン1.0～3.1がアプリケーションプログラムとして列記されている。

【0027】

また、閉じた状態とはなっているが、データベースプログラムとワードプロセッサプログラムも同様にフォルダ内に複数のアプリケーションプログラムを有している。

【0028】

一方、モジュールリスト表示窓43には、モジュールとしてのターゲット文書44（図1ではターゲット文書1）が複数表示されるようになっている。

ここで、ユーザーは表示装置（CRT）を参照して、どの文書（モジュール）をどのバージョンのアプリケーションプログラムで開くかを任意に決定することができる。

【0029】

同図において、たとえばターゲット文書44（モジュール）をスプレッドシートのバージョン2.0のアプリケーションプログラムで開きたい場合、マウス23によって画面上のマウスカーソル46に配置し、ここで一旦マウス29のボタン（図示せず）を押す。この行為（ドラッグ）によって、ターゲット文書44（モジュール）が選択されたことになり、アプリケーションリスト表示窓42では、このターゲット文書44を開くことのできるバージョンのアプリケーションプログラムのみが高い輝度表示で表示される。すなわち、当該ターゲット文書44を開くことのできるアプリケーションプログラムのバージョンが複数ある場合には、該当する複数のバージョンの全てが高輝度で表示される。

【0030】

次に、ユーザーは、前記マウス29のボタンを押した状態（ドラッグ状態）のままマウスカーソル46をアプリケーション表示窓42の任意のバージョン（ここではバージョン2.0）のアプリケーションプログラム上に移動させて、前記ボタンを開放する（ドラッグ）ことによってスプレッドシートのバージョン2.

0 のアプリケーションプログラムがターゲット文書 4 4（モジュール）を対象ファイルとして起動される。この手法は、いわゆる「ドラッグ・アンド・ドロップ」と呼ばれる画面インターフェース上の入力手法である。

【 0 0 3 1 】

図 4 で説明したインターフェースを実現するための機能構成を示したものが図 5 である。

同図において、文書解析制御部 5 1 は、図 1 における構文解析処理 2 および比較参照処理 3 を行うとともに、表示装置（C R T）2 5 への表示制御を行う機能を有している。すなわち、文書解析制御部 5 1 は、所定のフォルダまたはディレクトリ配下にあるターゲット文書 1 を順次読み込んでそのバージョンの解析を行い、モジュールリスト表示窓 4 3 にリスト形式で表示する。このモジュールリスト表示窓 4 3 に表示されたモジュール（ターゲット文書 4 4）は、ハードディスク装置（H D）2 7 内にインストールされたアプリケーションプログラムで読み込むことのできる正当な文書ファイルであるか否か、正当な文書ファイルである場合にはどのバージョンで開くことが可能なのかが文書解析制御部 5 1 によって既に識別された状態となっている。

【 0 0 3 2 】

文書解析制御部 5 1 は、ターゲット文書 1 を読み込んで解析を行った際に、この解析結果であるアプリケーション名とバージョン名とをアプリケーションパス取得部 5 4 に通知する。このアプリケーションパス取得部 5 4 はアプリケーションリスト表示窓 4 2 の表示を制御しており、文書解析制御部 5 1 から受け取ったバージョンのアプリケーションのみをアプリケーションリスト表示窓 5 4 に表示させる。

【 0 0 3 3 】

また、モジュールリスト表示窓 4 3 のターゲット文書 4 4（モジュール）がマウス 2 9 のドラッグ操作で前述のように選択された際には、この選択情報が文書解析制御部 5 1 を通じてアプリケーションパス取得部 5 4 に通知され、アプリケーションリスト表示窓 4 2 中の対応するアプリケーションプログラムのバージョンが高輝度表示されるようになっている。

【 0 0 3 4 】

そして、マウス 2 9 のドロップ操作で所定のバージョンのアプリケーション（ここではバージョン 2. 0 のスプレッドシートのアプリケーションプログラム 4 5）が選択されると、この選択情報がアプリケーションパス取得部 5 4 を通じて実行パスとして O S (Operating System) に通知される。

【 0 0 3 5 】

これにより O S は、アプリケーションパス取得部 5 4 より実行ファイルパスを、文書解析制御部 5 1 より文書ファイルパスを受け取り、対象となるターゲット文書 4 4 （モジュール）を選択されたバージョンのアプリケーションプログラムで実行することになる。

【 0 0 3 6 】

なお、本実施例において O S としては、マイクロソフト社のウィンドウズ 3. 1、9 5 または 9 8 等を用いることができる。

図 6 および図 7 は比較画面 4 1 の表示において、ターゲット文書（モジュール）と対応するアプリケーションプログラムのバージョンとの対比表示の変形例である。

【 0 0 3 7 】

図 6 は、アプリケーションプログラムのバージョン毎に異なるアイコンを定義し、ターゲット文書 4 4 （モジュール）を表示する場合に対応するバージョンのアイコンで表示させるようにしたものである。

【 0 0 3 8 】

この点について、既存の O S では、ファイル名の拡張子（たとえば X L S や D O C）によってアイコンを定義しているため、同一の拡張子が付与されているとバージョンが異なっても同じアイコンで表示されてしまうため、アイコンを目視しただけでは当該ターゲット文書のバージョンの把握が困難だった。

【 0 0 3 9 】

この点について図 6 に示した例では、文書解析制御部 5 1 において全てのターゲット文書 4 4 についてその対応バージョンが把握されているため、ファイル名の拡張子に左右されることなくターゲット文書の対応バージョンがアイコンによ

って一目で識別可能となるように表示できる。

【 0 0 4 0 】

図 7 は、アプリケーションリスト表示窓 4 2 で指定したバージョンに対応したターゲット文書 4 4 のみをモジュールリスト表示窓 4 3 に表示するようにした例である。この場合には、アプリケーションリスト 5 3 で特定のバージョンが指定されると、アプリケーションパス取得部 5 4 がこれを検出し、文書解析制御部 5 1 に当該バージョンが指定されていることを通知する。この通知に基づいて文書解析制御部 5 1 では前記バージョンに対応したターゲット文書（モジュール 4 4）のみをモジュールリスト表示窓 4 3 に表示させることができる。

【 0 0 4 1 】

図 8 は、ターゲット文書によってアプリケーションプログラム 5 3 のバージョンを選択する概念を示している。図 4，図 6，図 7 で説明した画面インターフェースでターゲット文書 8 4 a，8 4 b が文書解析制御部 5 1 に読み込まれると、これが解析詳細部 5 1 a で解析されてインストール済みのアプリケーションプログラムの中から最適なバージョンのアプリケーションプログラム 4 5 a を選択するようになっている（図 8 の上図）。そして、解析詳細部 5 1 a で決定されたアプリケーションプログラム 4 5 a によってターゲット文書 8 4 a が開かれるようになっている。

【 0 0 4 2 】

図 9 は、文書解析制御部 5 1 による解析の結果、アプリケーションリスト 5 3 内に該当するバージョンのアプリケーションプログラムが存在しない場合の機能ブロック構成を示している。同図において大半の構成は図 5 と同様であるのでその部分の説明は省略する。

【 0 0 4 3 】

インストール制御部 9 1 は、ハードディスク装置（HD）2 7 内にアプリケーションプログラムを実行可能な状態にインストールするための制御部であり、具体的には端末装置 2 1 の図示しない CD-ROM やサーバ 3 1 からアプリケーションプログラムをハードディスク装置（HD）2 7 内にインストールする機能を有している。すなわち、図 9 では、アプリケーションリスト 5 3 からアプリケー

ションパス取得部54がパスを取得できない場合、つまり、モジュールリスト52で選択されたターゲット文書44（モジュール）に対応するバージョンが存在しない場合に、インストール制御部91が起動されるようになっている。

【0044】

図10は、インストール制御部91による新たなバージョンのアプリケーションプログラムのインストール手順を示したものである。

同図では、バージョンnのターゲット文書1001, 1002が順次文書解析制御部51に読み込まれ、解析詳細部51aでアプリケーションプログラムのバージョンを判定したにもかかわらず、インストール済みのアプリケーションプログラム（1003）はバージョンmであり、前記バージョンnのターゲット文書1001, 1002に対応していない状態となっている（図10（a））。

【0045】

このとき、文書解析制御部51からの通知を受けて、インストール制御部91が起動し、ターゲット文書1001, 1002のバージョンnに対応したアプリケーションプログラム1004が新たにインストールされる。（図10（b））

【0046】

ここで、当該アプリケーションプログラムのインストール元は、端末装置21内のハードディスク装置（HD）27であってもよいし、図示しないCD-ROMドライブ装置からであってもよいし、さらに、ネットワーク30を介したサーバ31であってもよい。ネットワーク30を介してサーバ31からアプリケーションプログラムをダウンロードする場合には、TCP/IPによるFTP（File Transfer Protocol）を用いることができる。

【0047】

ハードディスク装置（HD）27内に新たにインストールされたアプリケーションプログラム1004が起動して、これに対応したバージョンのターゲット文書1001または1002が開かれる（図10（c））。

【0048】

図11および図12は、必要なバージョンのアプリケーションプログラムをイ

インストールする際に、インストール領域を確保するための機能ブロック図、その処理手順を示す図である。

【 0 0 4 9 】

図 1 1 の構成は図 9 で示した構成にアンインストール制御部 1 1 0 1 が付加されている点が特徴である。すなわち、図 9 および図 1 0 で説明した例でインストール制御部 9 1 が起動されたにもかかわらず、ハードディスク装置 (H D) 2 7 内に新たなバージョンのアプリケーションプログラムをインストールする領域が不足している場合がある。

【 0 0 5 0 】

この場合には、図 1 1 で示すアンインストール制御部 1 1 0 1 が起動される。このアンインストール制御部 1 1 0 1 もプログラムとして実現されており、中央処理装置 (C P U) 2 3 が当該プログラムを実行することにより具体化される。

【 0 0 5 1 】

図 1 2 は、上記アンインストールおよびインストールの処理手順を示している。

解析詳細部 5 1 a の解析によってインストール済みのアプリケーションプログラムから最適なバージョンのものが検出できなかった場合 (図 1 2 (a))、インストール制御部 9 1 が起動するが、ハードディスク装置 (H D) 2 7 に新たなアプリケーションプログラムをインストールする領域が残っていない場合がある (図 1 2 (b))。

【 0 0 5 2 】

このような場合に、インストール制御部 9 1 はアンインストール制御部 1 1 0 1 に対して領域不足を通知する。これによって、アンインストール制御部 9 1 はハードディスク装置 (H D) 2 7 中から不要なファイルを削除する。このときにいずれのファイルを削除するか否かについては、下記のような基準で決定することができる。

【 0 0 5 3 】

たとえば、利用条件を満たしていないアプリケーションプログラムを優先的にアンインストールするようにしてもよい。たとえば、ダウンロード後 3 0 日間の

み試用可能な条件が付加されているようなアプリケーションプログラムがこれに該当する。

【 0 0 5 4 】

また、容量順にアンインストールすることも可能である。一般に、容量が少ないアプリケーションほど重要度が低い傾向がある場合、容量が小さなアプリケーションプログラムから先に削除する。また、アンインストールするプログラム本数を少なくして領域確保効率を優先させるのであれば、容量が大きなアプリケーションプログラムから削除する。

【 0 0 5 5 】

また、インストールした日付順、重要度、作成文書数、共通ファイルの数の大小等によってアンインストールする順番を決定してもよい。

図 1 3 および図 1 4 は、ネットワーク 3 0 を通じて端末装置 2 1 がサーバ 3 1 からアプリケーションプログラムをダウンロードする手順を示すシーケンス図である。

【 0 0 5 6 】

同図においてまず、サーバ 3 1 は管理プログラムを、端末装置 2 1 はクライアント用プログラムをそれぞれ起動する。なお、端末装置 2 1 のハードディスク装置 (HD) 2 7 内には、端末装置 2 1 がサーバ 3 1 から取得したファイルの履歴がファイル取得履歴ファイル 1 3 0 1 として格納されている。

【 0 0 5 7 】

端末装置 2 1 のインストール制御部 9 1 によりインストール処理が開始されると、まず端末装置 2 1 よりサーバ 3 1 に対して利用開始メッセージが送信される。これにより、サーバ 3 1 がインストール開始を認識すると、自身の自動管理システムが有効であることを端末装置 2 1 に通知する。

【 0 0 5 8 】

次に、端末装置 2 1 は、利用者識別符号 (ID) をサーバ 3 1 に通知する。これを受領したサーバ 3 1 は、端末装置 2 1 に対してハードウェア情報を要求する。このハードウェア情報とは、端末装置 2 1 の中央処理装置 (CPU) 2 3 の種別、クロック周波数 (Clock)、OS、メモリ容量、ハードディスク装置 (HD

） 2 7 の容量等である。これらの情報を受け取ると、次に端末装置 2 1 に対してファイル取得履歴を要求する。このファイル取得履歴とは、いわゆるヒストリであり、ファイル名、URL (Uniform Resource Locator)、圧縮方式、バージョン等の情報であり、前述のファイル取得履歴ファイル 1 3 0 1 として端末装置 2 1 がハードディスク装置 (HD) 2 7 が有しているものである。

【 0 0 5 9 】

サーバ 3 1 がこのファイル取得履歴を受領すると、サーバはこれに基づいて過去にダウンロードしたファイルの新しいバージョンが既にサーバにダウンロード可能な状態で格納されているか否かを判定する。そして、新しいバージョンが存在しない場合には、端末装置 2 1 にはサーバ 3 1 からみて最新のバージョンのアプリケーションプログラムが既にインストール済みであるため、処理を終了する。

【 0 0 6 0 】

一方、端末装置 2 1 から得たファイル取得履歴よりもさらに新しいバージョンがサーバ 3 1 側に存在する場合には、必要なインストーラ (インストールを行うための補助プログラム) を選択し、これを端末装置 2 1 に転送する。

【 0 0 6 1 】

端末装置 2 1 からサーバ 3 1 に前記インストーラの転送終了が通知されると、サーバ 3 1 は端末装置 2 1 に対してファイル取得履歴の書き換えを命令する。端末装置 2 1 はこの命令に基づいて、サーバ 3 1 にバージョンやファイル日付等の必要項目を要求し、サーバ 3 1 はこれに応答して前記項目の情報を端末装置 2 1 に通知する。

【 0 0 6 2 】

端末装置 2 1 では、前記項目情報に基づいて自身のファイル取得履歴ファイル 1 3 0 1 を書き換えるとともに、サーバ 3 1 とのネットワークを切断する。

そして、端末装置 2 1 では、ダウンロードしたインストーラを起動してインストールを開始する。ここでインストールが正常終了しなかった場合には、アーカイブの正当性検査を開始する。このアーカイブ正当性検査は、再度サーバ 3 1 とのネットワーク接続を行い、サーバ 3 1 に対して検査要求を送信することによ

て行われる。この検査結果に基づいて必要な場合には前記インストーラを再度サーバ 3 1 から端末装置 2 1 に転送する。

【 0 0 6 3 】

【発明の効果】

本発明によれば、ファイルの外観からは特定できないデータファイルに対して、最適なバージョンのアプリケーションプログラムを決定することができるとともに、当該アプリケーションプログラムを自動インストールし、当該アプリケーションの自動起動を行うことができる。

【図面の簡単な説明】

【図 1】 本発明の実施例による自動インストールシステムの機能ブロック図

【図 2】 実施例 1 のシステム構成を示すブロック図

【図 3】 実施例 1 の各リストの関係を示すブロック図

【図 4】 実施例 2 の表示装置における画面表示例

【図 5】 図 4 の画面構成を実現するための機能ブロック図

【図 6】 実施例 2 の表示装置における画面表示例（2）

【図 7】 実施例 2 の表示装置における画面表示例（3）

【図 8】 実施例 2 の文書ファイルとアプリケーションプログラムとの関連を示す機能ブロック図

【図 9】 実施例 2 において、文書解析制御部による解析の結果、アプリケーションリスト内に該当するバージョンのアプリケーションプログラムが存在しない場合の機能ブロック図

【図 1 0】 実施例 2 において、文書解析結果に基づいてアプリケーションプログラムをインストールする場合の機能ブロック図

【図 1 1】 実施例 2 において、インストール領域が不足している場合のアンインストールを行う場合の機能ブロック図

【図 1 2】 実施例 2 において、アンインストールを実行して領域を確保する手順を説明する機能ブロック図

【図 1 3】 実施例 2 において、ネットワークを介してアプリケーションプログラムをダウンロードする場合のシーケンス図（1）

【図 1 4】 実施例 2 において、ネットワークを介してアプリケーションプログラムをダウンロードする場合のシーケンス図（2）

【符号の説明】

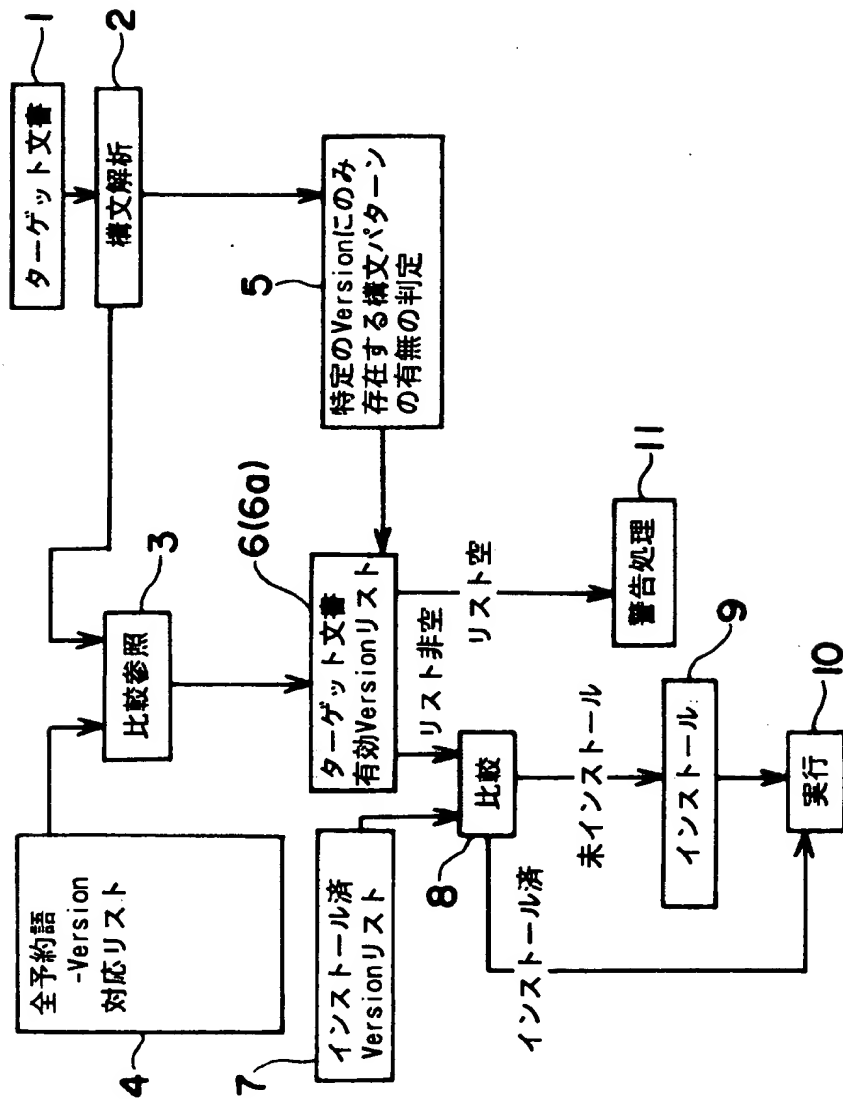
- 1 ターゲット文書
- 4 全予約語－バージョン対応リスト
- 6 ターゲット文書有効バージョンリスト
- 7 インストール済みバージョンリスト
- 2 1 端末装置
- 2 2 バス
- 2 3 中央処理装置（CPU）
- 2 4 メモリ
- 2 5 表示装置（CRT）
- 2 6 印刷装置（PRT）
- 2 7 ハードディスク装置（HD）
- 2 8 通信インターフェース
- 2 9 マウス
- 3 0 ネットワーク
- 3 1 サーバ
- 4 1 比較画面
- 4 2 アプリケーションリスト表示窓
- 4 3 モジュールリスト表示窓
- 4 6 マウスカーソル

【書類名】

図面

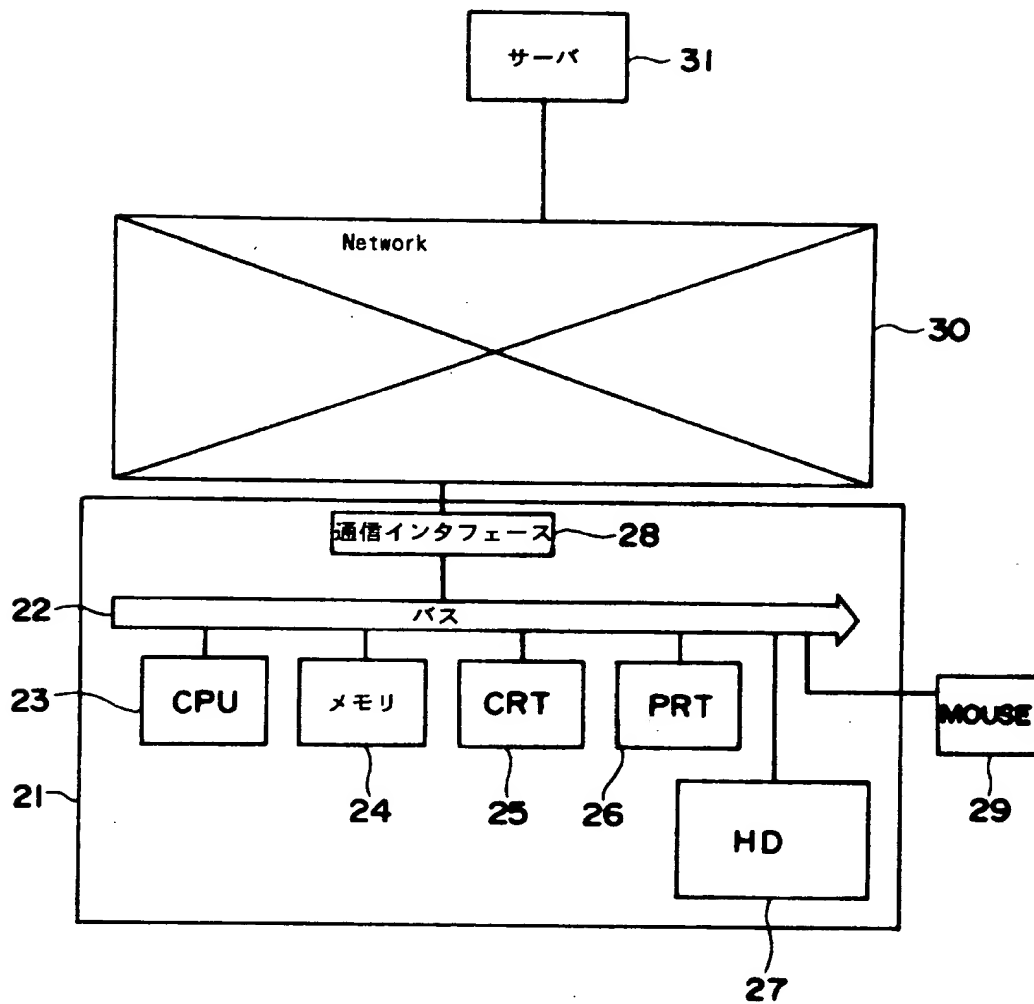
【図 1】

本発明の実施例による自動インストールシステムの機能ブロック図



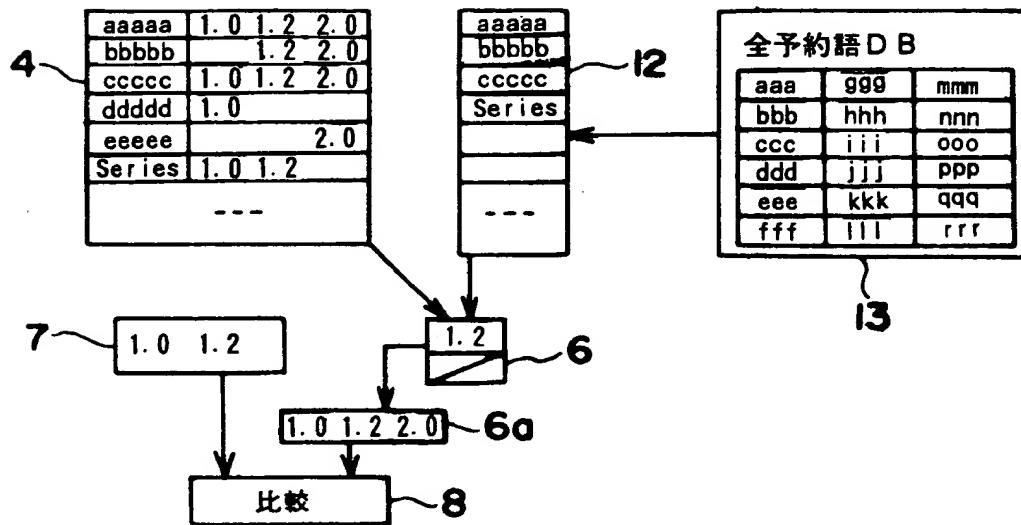
【図 2】

実施例 1 のシステム構成を示すブロック図



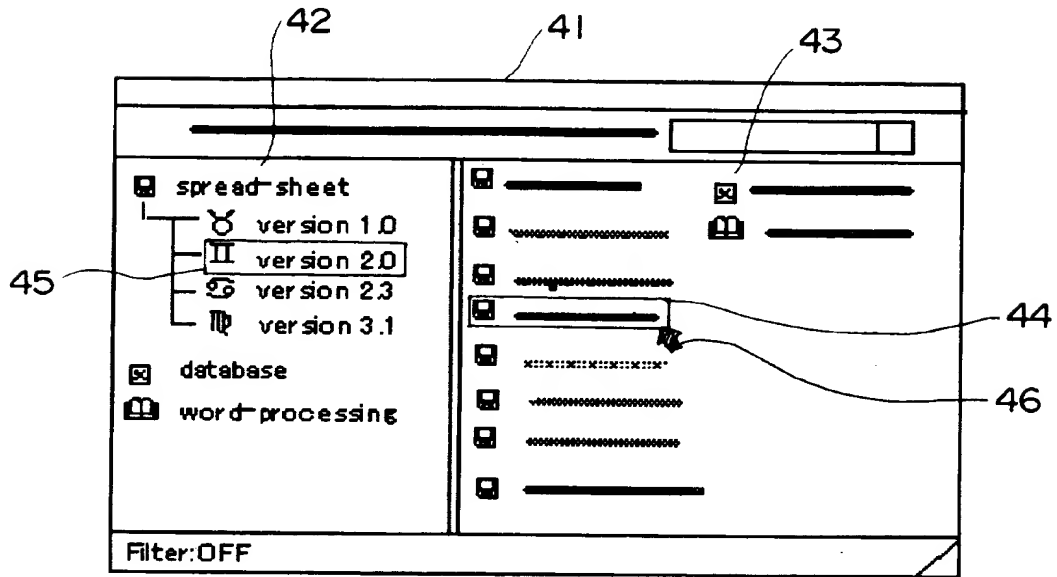
【図 3】

実施例 1 の各リストの関係を示すブロック図



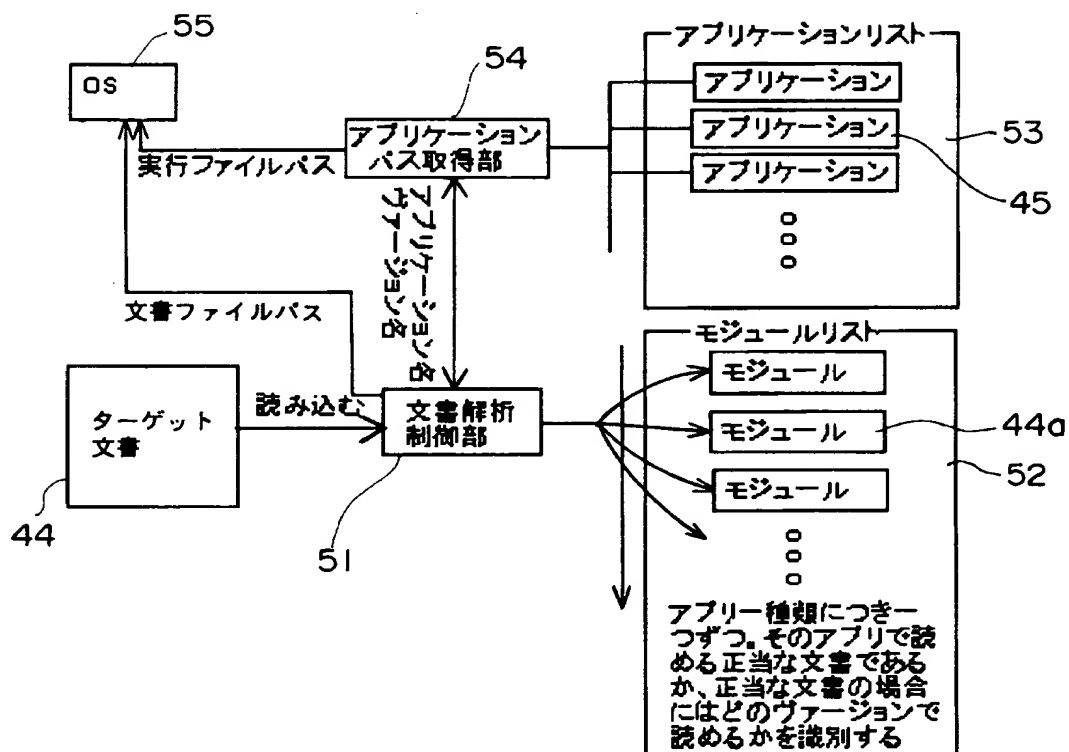
【図 4】

実施例 2 の表示装置における画面表示例



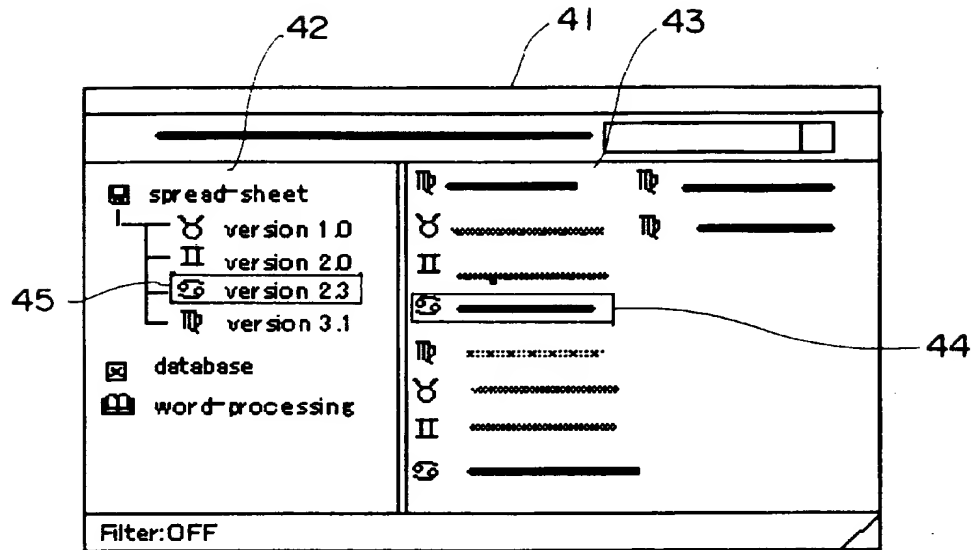
【図 5】

図 4 の画面構成を実現するための機能ブロック図



【図 6】

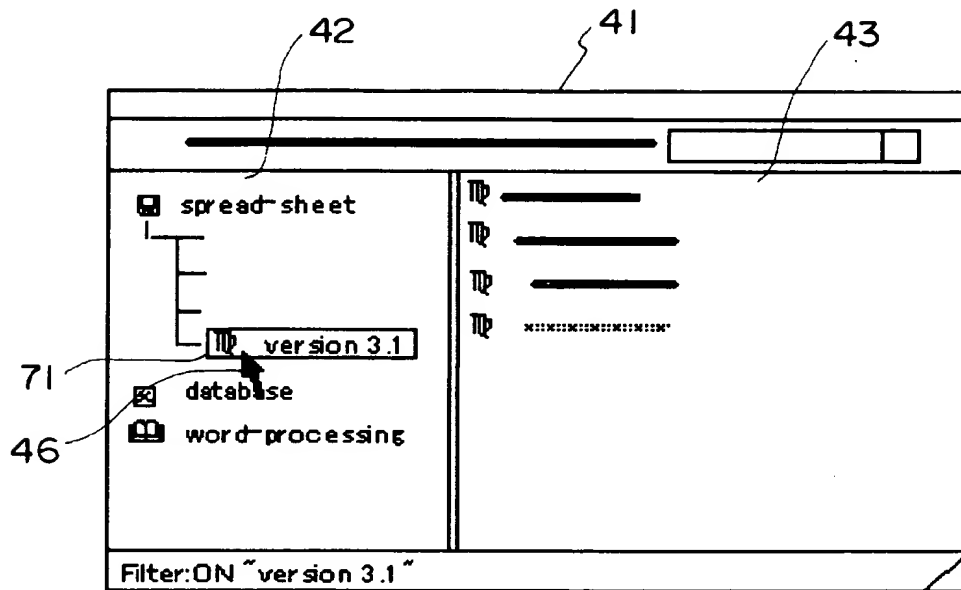
実施例 2 の表示装置における画面表示例 (2)



通常同一のアプリケーションにより作成された文書はバージョンによらず同一のアイコンを伴うことが多いが、たとえば図に示すようにバージョンによりその文書アイコンを意図的に変えて表示することにより、利用者に対する利便がはかられる。

【図 7】

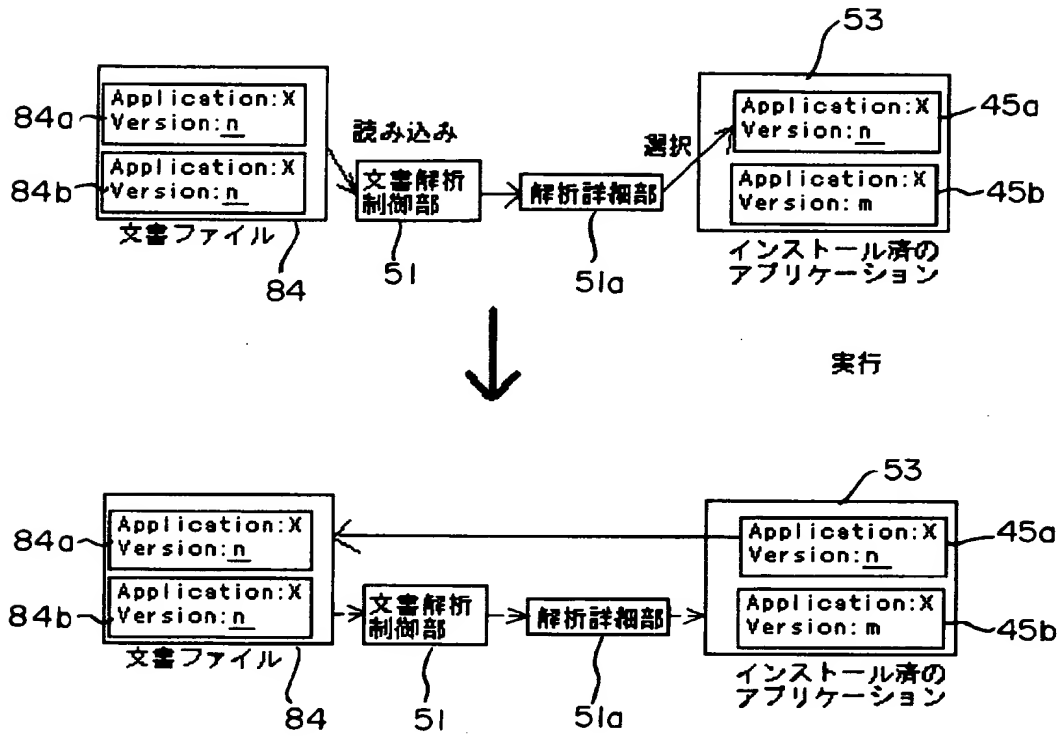
実施例 2 の表示装置における画面表示例 (3)



また文書ファイルのリストを表示する際に、特定のバージョンのアプリケーションにより読み取ることが出来る文書の名前のみを表示させることにより、利用者がいちいちアイコンの違いを識別する必要がなくなり、さらに利便がはかれることとなる。

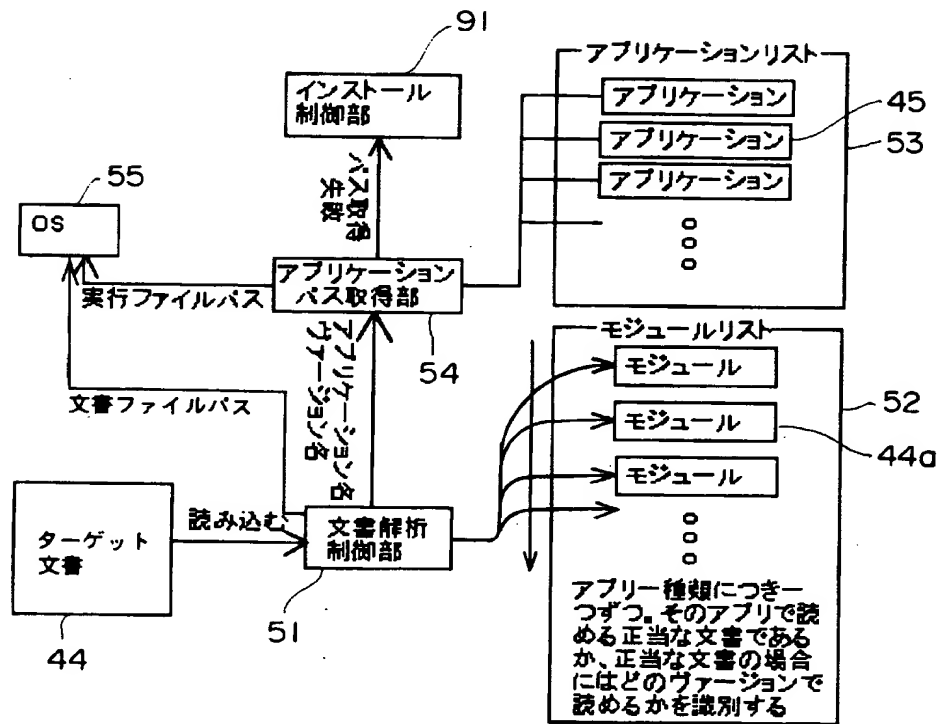
【図 8】

実施例 2 の文書ファイルとアプリケーションプログラムとの
関連を示す機能ブロック図



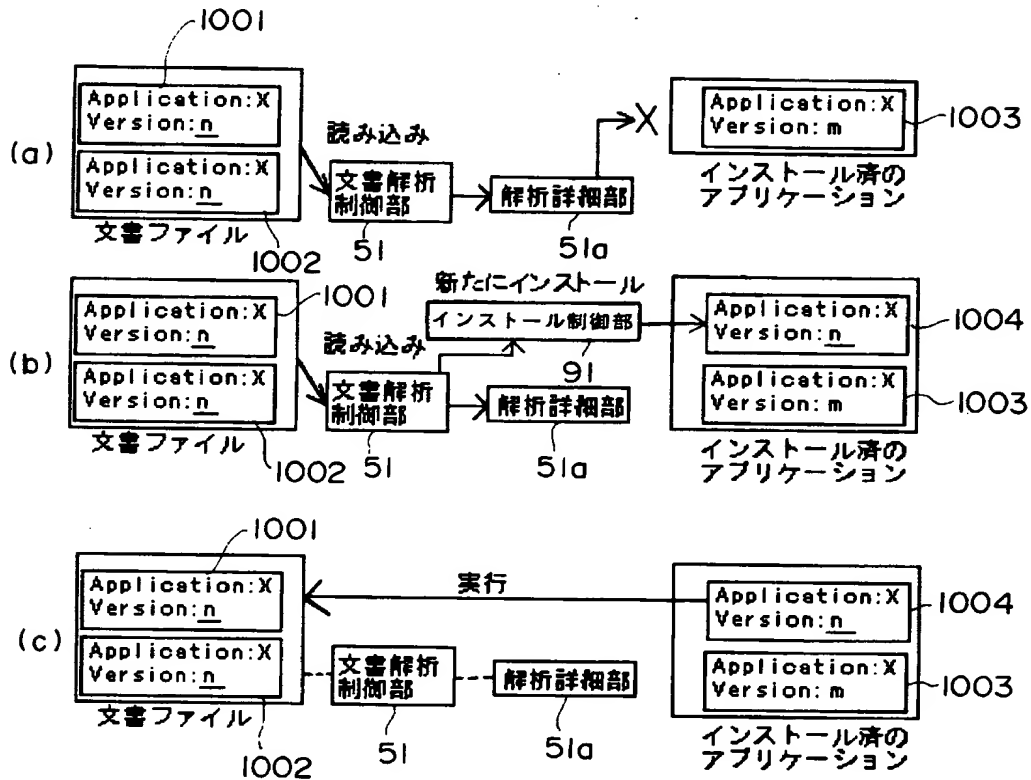
【図 9】

実施例 2 において、文書解析制御部による解析の結果、アプリケーションリスト内に該当するバージョンのアプリケーションプログラムが存在しない場合の機能ブロック図



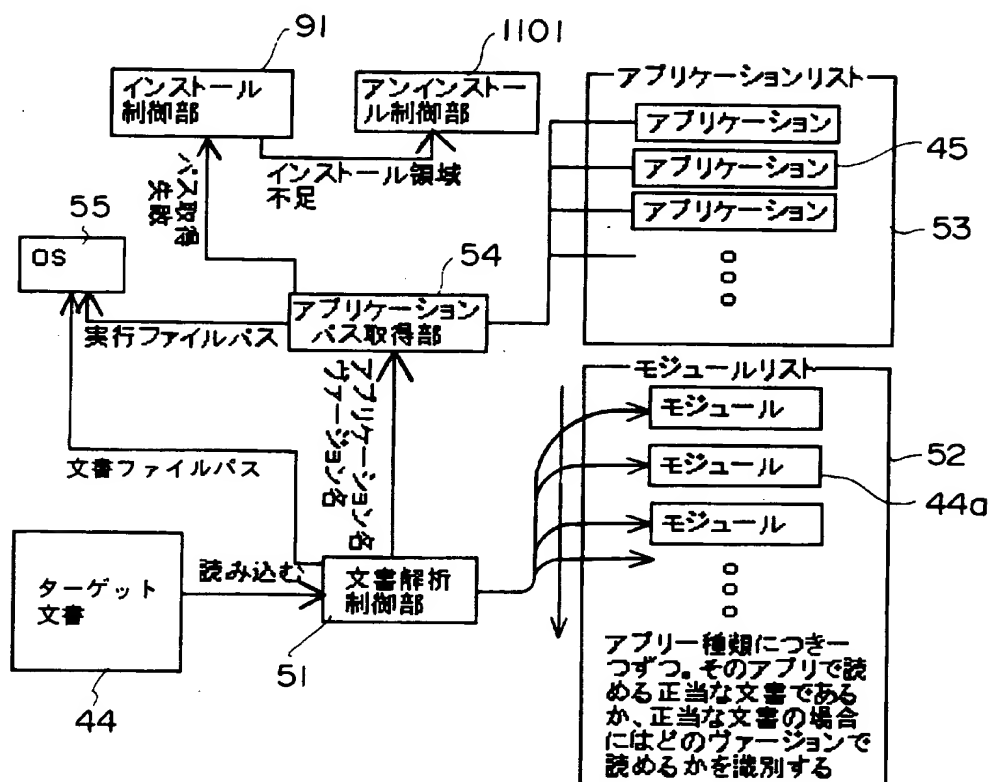
【図 10】

実施例 2 において、文書解析結果に基づいてアプリケーションプログラムをインストールする場合の機能ブロック図



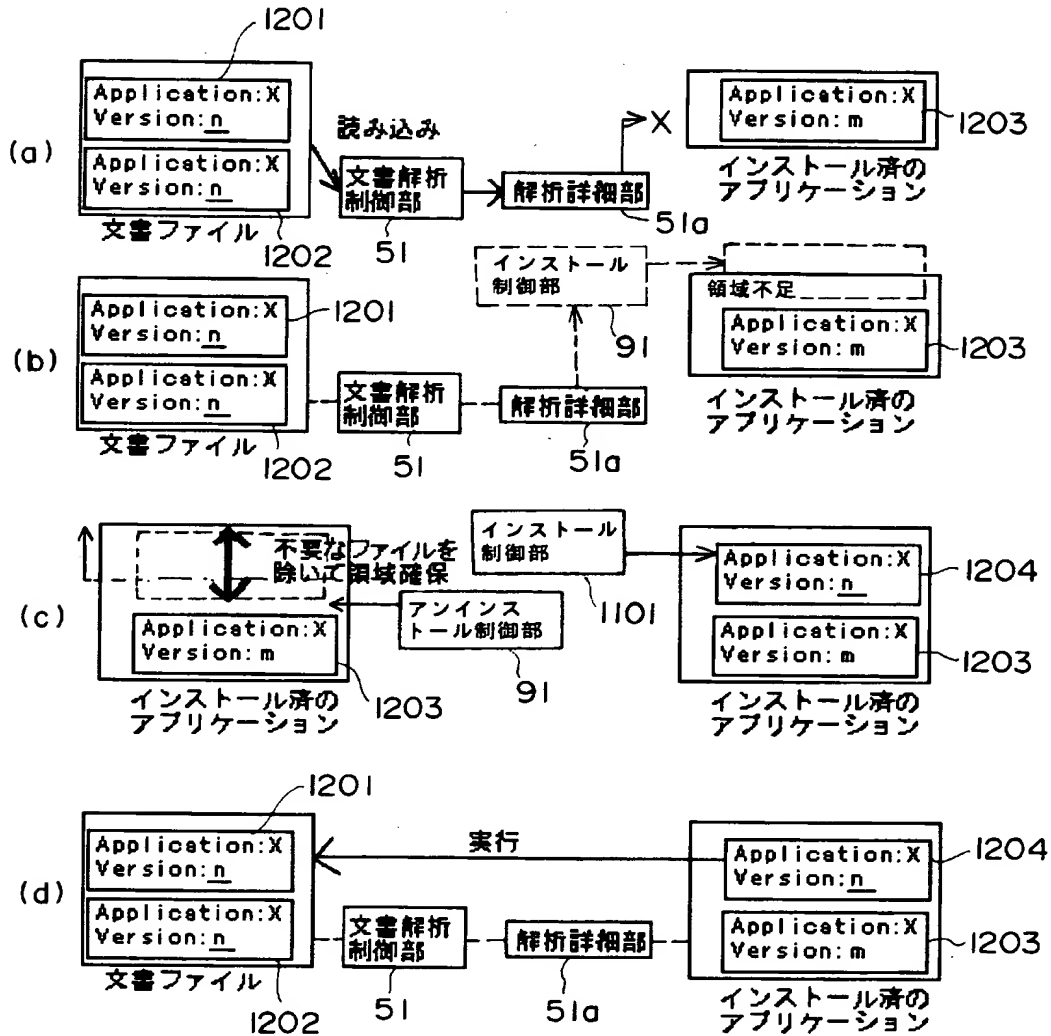
【图 1-1】

実施例 2 において、インストール領域が不足している場合のアンインストールを行う場合の機能ブロック図



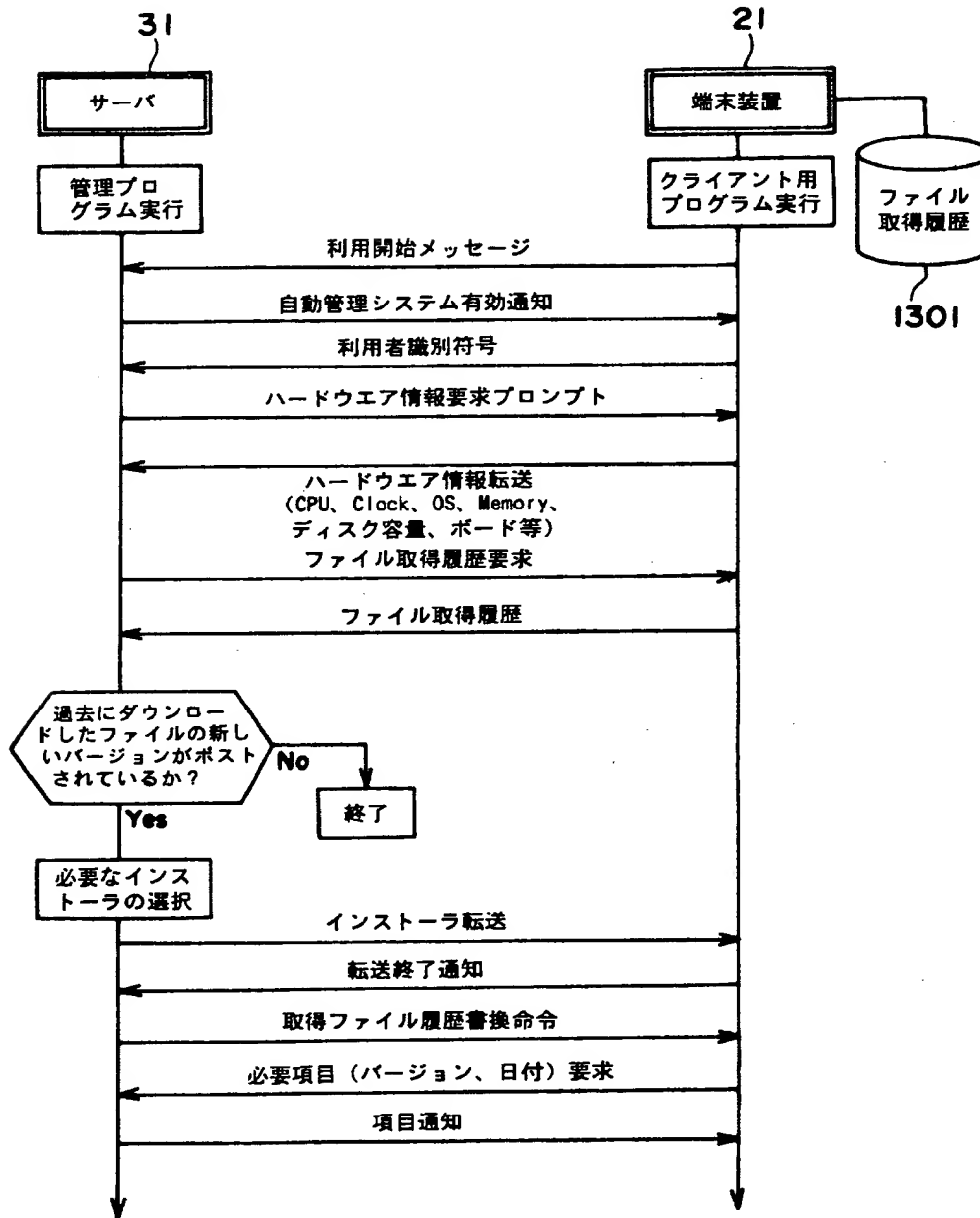
【図 1 2】

実施例 2 において、アンインストールを実行して領域を確保する手順を説明する機能ブロック図



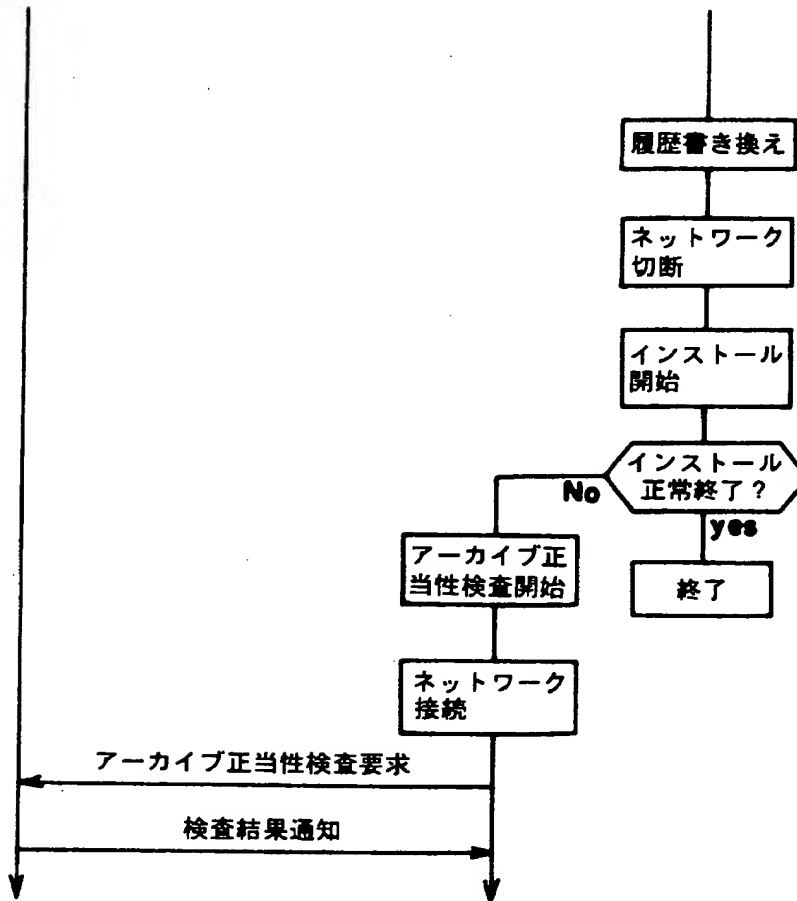
【図 1 3】

実施例 2 において、ネットワークを介してアプリケーションプログラムをダウンロードする場合のシーケンス図 (1)



【図 14】

実施例 2 において、ネットワークを介してアプリケーションプログラムをダウンロードする場合のシーケンス図 (2)



【書類名】 要約書

【要約】

【課題】 ユーザーにわずらわしいバージョンの判定を行わせることなく、データファイルに対して最適なバージョンのアプリケーションプログラムを自動インストールし、当該アプリケーションの自動起動を可能にする。

【解決手段】 読み出そうとする文書ファイルを解析してマクロ命令の予約語等のような特徴点を抽出し、抽出された特徴点によってその文書ファイルに適したアプリケーションのバージョンを識別する。そして、前記で識別されたバージョンのアプリケーションプログラムが既にインストールされているか否かを判定し、この判定結果によりインストールされていないときには当該バージョンのアプリケーションプログラムを新たにインストールするようにした。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [000005223]

1. 変更年月日	1996年 3月26日
[変更理由]	住所変更
住 所	神奈川県川崎市中原区上小田中4丁目1番1号
氏 名	富士通株式会社